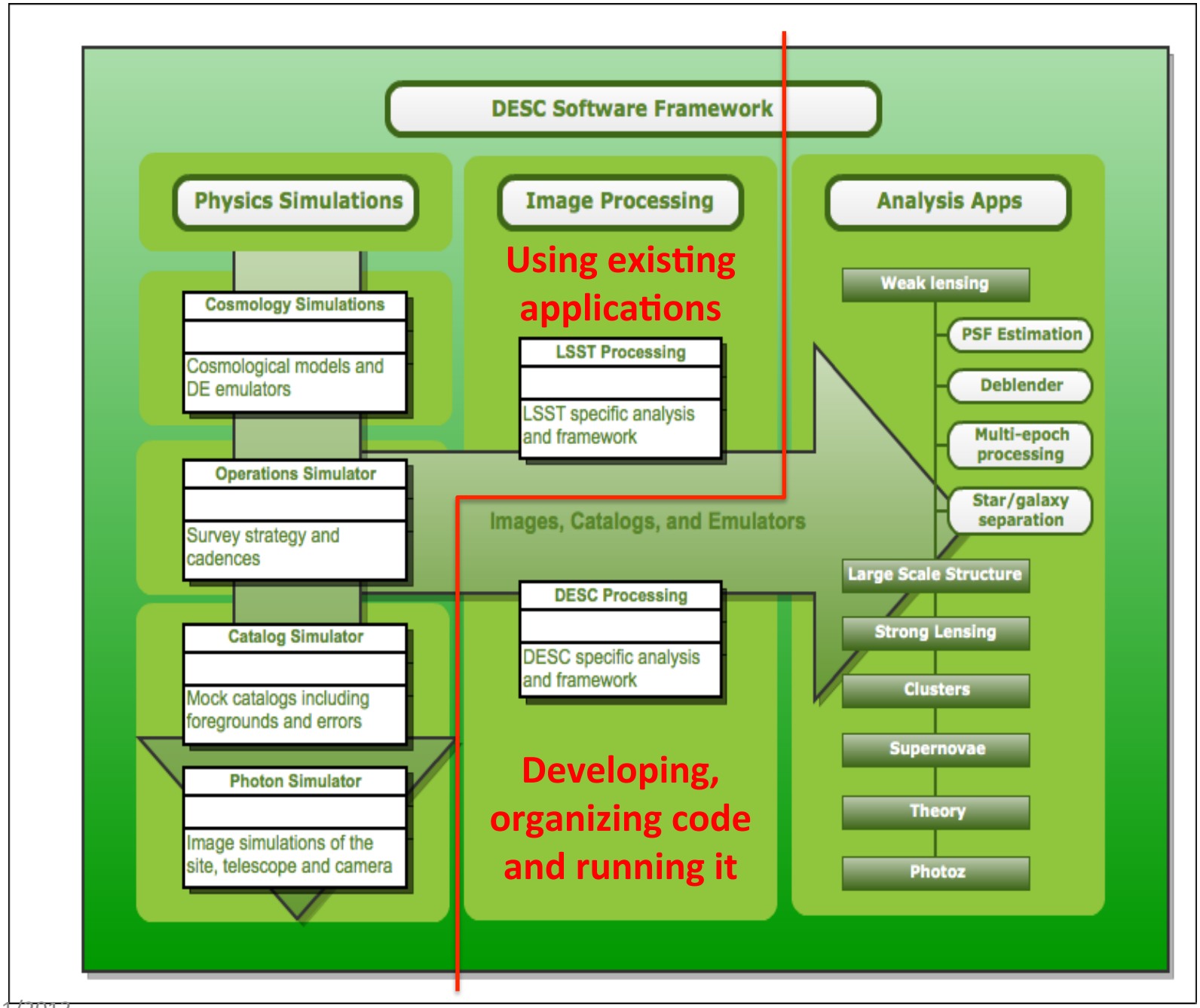
Tools to construct a DESC Analysis Framework

Jim Kowalkowski

Fermilab Scientific Computing

Goals

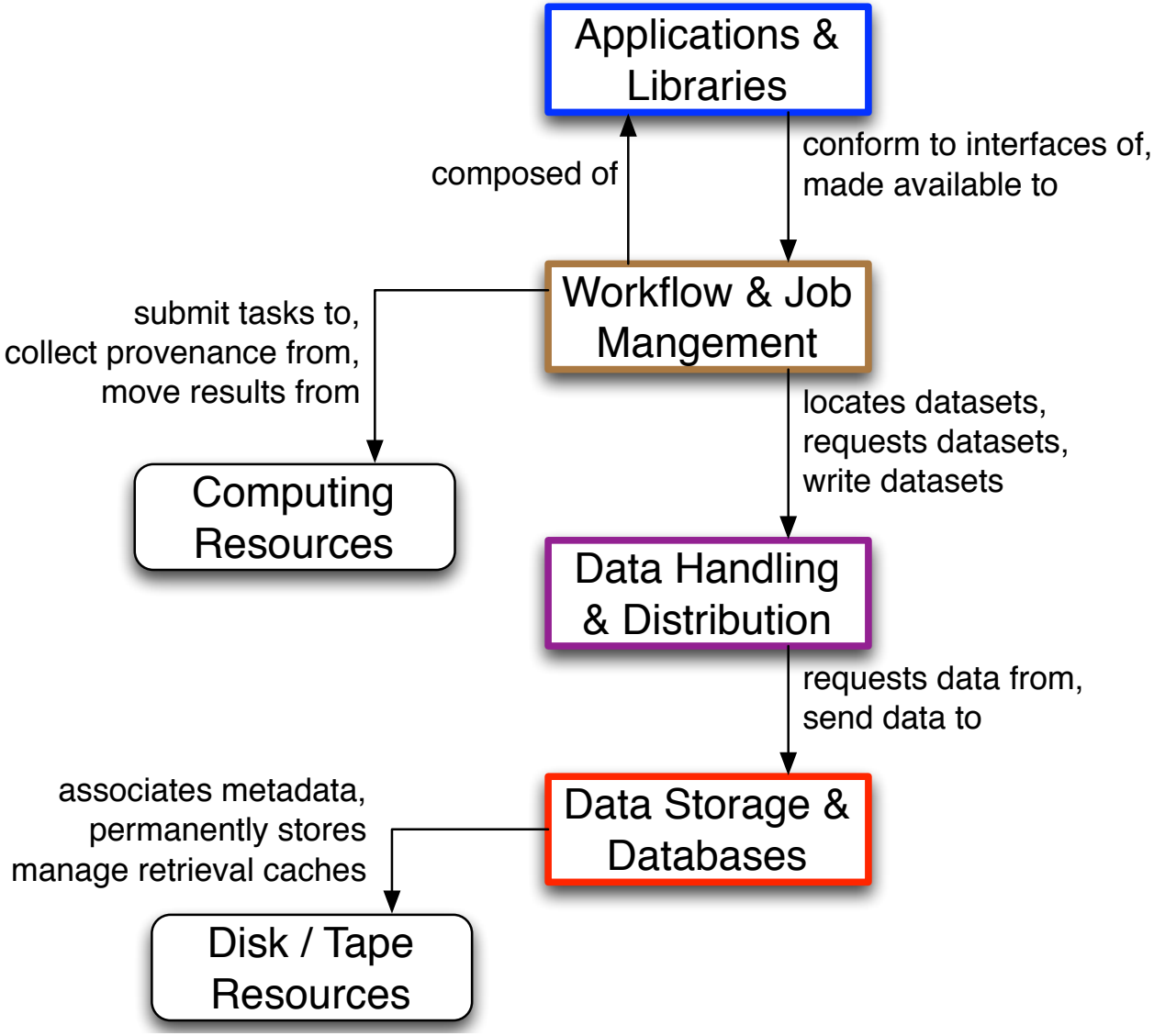
- Achieve common understanding of what we are constructing
  - Describe desirable features for a DESC software framework
  - Present a possible view and description of its software components
- Discuss some of the tools FNAL uses to construct systems
  - Show the software products used within each component for experiments that we work with at FNAL
  - Explain how these products are used in the context of DES as an example for similar science processing tasks (Brian and Steve will cover this).
- Discuss how we might proceed and some of FNAL strengths


DESC Software Framework
Physics Simulations
Image Processing
Analysis Apps
Using existing applications
Cosmology Simulations
Cosmological models and DE emulators
LSST Processing
LSST specific analysis and framework
Operations Simulator
Survey strategy and cadences
Images, Catalogs, and Emulators
DESC Processing
DESC specific analysis and framework
Catalog Simulator
Mock catalogs including foregrounds and errors
Photon Simulator
Image simulations of the site, telescope and camera
Developing, organizing code and running it
Weak lensing
PSF Estimation
Deblender
Multi-epoch processing
Star/galaxy separation
Large Scale Structure
Strong Lensing
Clusters
Supernovae
Theory
Photoz

DESC software subsystems

- **Physics applications and libraries**: develop, test, organize, catalog, and find source and executables.
- **Job specification, submission, and management**: run application sequences (workflows) where it is efficient to get the work done and collect provenance and other metadata.
- **Data handling and distribution systems**: find datasets to be processed and make them available to a job wherever it is run, move metadata and results from jobs to permanent storage and catalog them.
- **Data storage and databases**: permanently store and archive datasets with associated metadata and permit timely access to them.

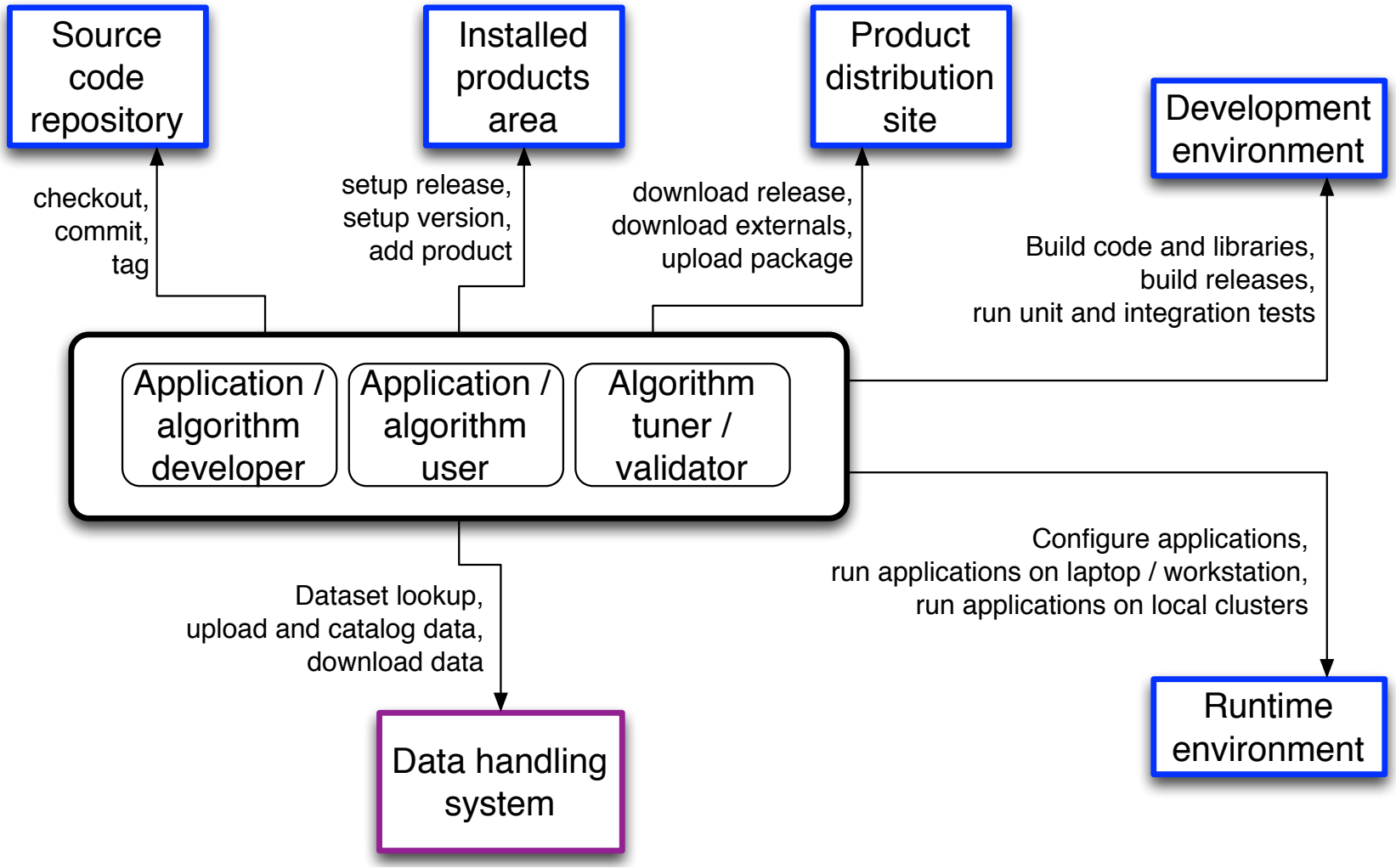
Main relationships amongst the pieces



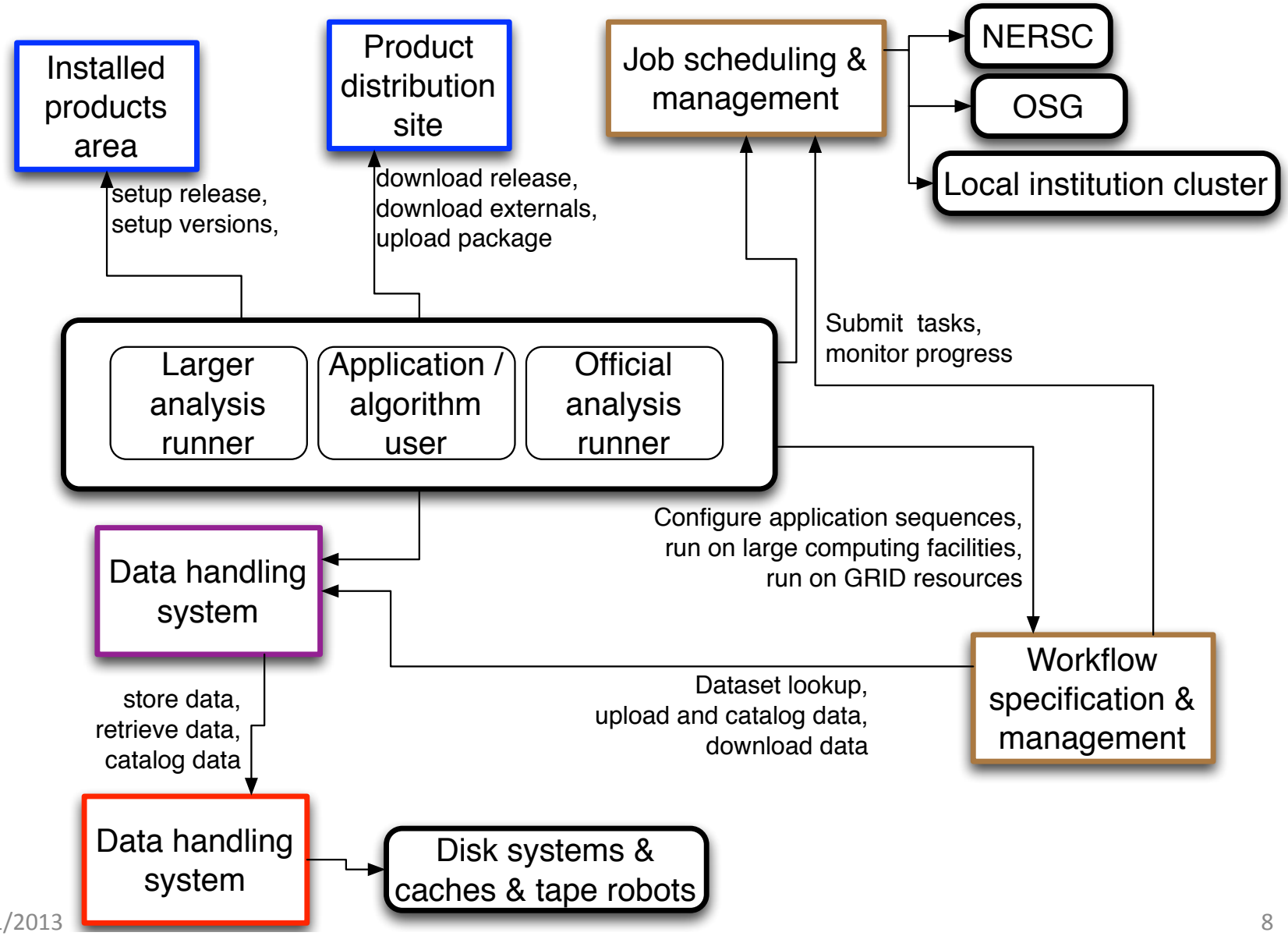
Some analysis framework user roles

- **Application / algorithm developer:**
  - Build, test, and make code available to others
- **Application / algorithm user**
  - Developers are often users of others' code
- **Algorithm tuner / validator**
  - May use applications outside the workflow system
- **Large-analysis runner**
  - Use of workflow system on local and GRID resources
- **Official-analysis runner**
  - Use of official releases on many resources

Local resource interactions



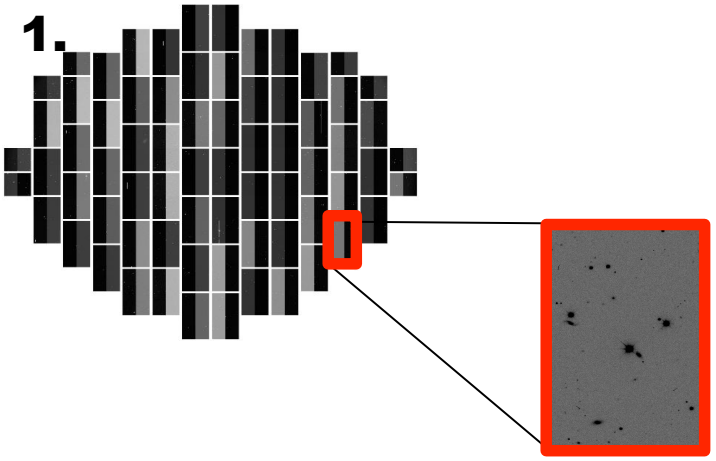
Larger analysis role interactions

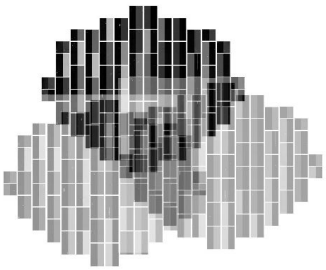


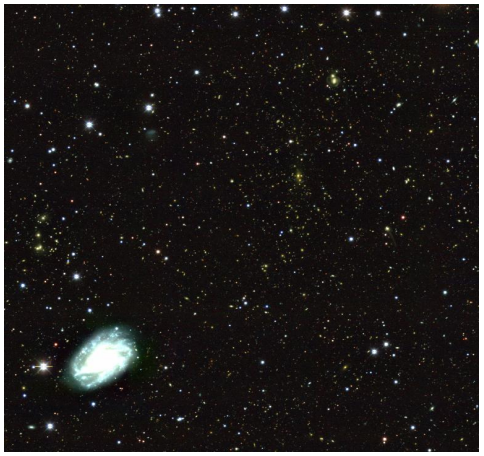
Technologies used by FNAL experiments

Applications	**Workflow**	**Data handling**	**Data storage**
Relocatable UPS	GlideinWMS	SAM-lite	Enstore
MessageFacility	Condor / PBS	GridFTP	dCache
Redmine + git	DAGMAN	(Globus Online)	SAM-lite
Art (IF framework)	(Galaxy)	(Galaxy)	(IRODS)
Buildtools / cmake	WMAgent (CMS)	Phedex (CMS)	Oracle RDBMS

- FNAL statistics
  - 94 Petabytes written to tape (Enstore), mostly from offsite
  - 160 Gbps LAN traffic from archive to local processing farms
  - LHC peak WAN usage into and out of Fermilab at 20-30 Gbps
- Tier-1 CMS of FNAL
  - near-line storage: 15 PB, with 60/2-3 TB disks
  - Tape: 22 PB (STK SL8500 robots)
  - Average of ~30K analysis jobs daily
- CMSSW framework usage
  - 3M lines & growing
  - >1000 DLLs, >400 external libraries
  - >1000 developers over lifetime, 100s at one time







Simulated and on-sky DECam data exposures → **Mosaic registration and calibration** → **multi-band coadd catalogs**

Active example using the DES Wrapper and Workflow tools integrated to astronomy codes:

1. Generate DES simulations, on sky DECam exposures.

2. Process on-sky DECam Data.

3. Analyze object catalogs for science results (photo-z).

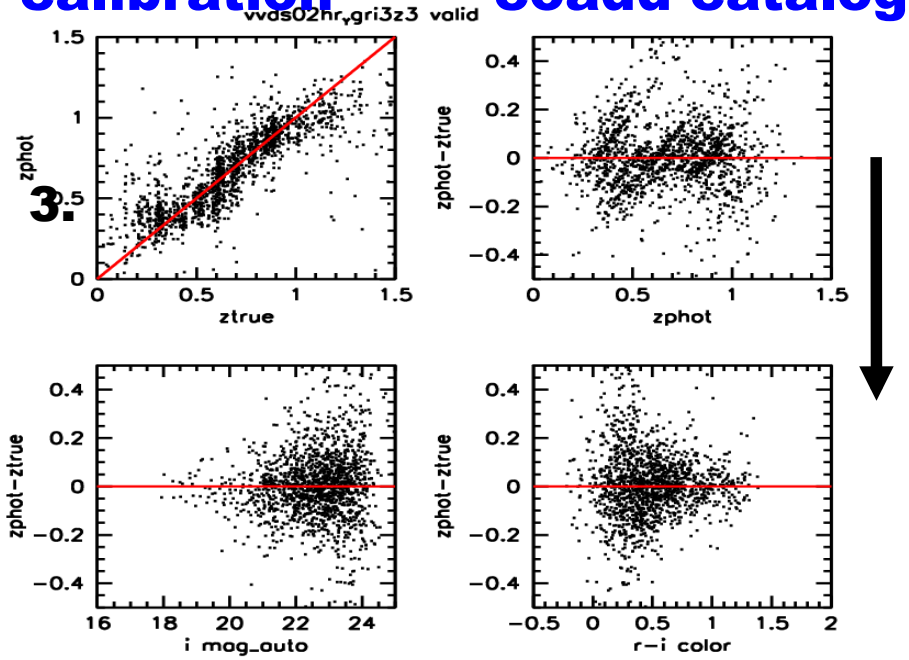


Photo-z science analysis

DES Wrappers and Workflow system

What is it: A set of rules and tools to enable running a set of heterogeneous astronomical data processing codes on a variety of platforms from a user's laptop to an Open Science Grid cluster, to a High Performance Computing site such NERSC or XSEDE. The system tracks software and parameter versions, image files, and catalog provenance, all optionally interfaced to a database.

Example rules:

File names are unique – proper namespace management

Wrappers provide uniform calling structure to all underlying codes – chain together large variety of codes

Provenance information (versions, file parentage) tracked across multiple reprocessings of data – allows development and bug tracking

Example tools:

Python wrapping code, WCL (XML-style parameter control language), EUPS (manage executables and dependencies), SVN(source code repository), Globus-online (remote file distribution), Condor (distributed remote job execution), dCache (Petabyte scale disk cache), ENSTORE (Petabyte scale tape backend)

One new element of the DES system is the 'wrapper+workflow' component.

Modules to be run are specified in a '.wcl' (XML-style) language, This specifies the input files, executable (with versions), parameter settings:

```
Running: ../DES_HOME/bin/generic.py --inp wrap_firstcut.wcl >& ./03_astrf/log/createscampcats-00139389_35.log
Running: ../DES_HOME/bin/genList.py --input wrap_firstcut.wcl >& ./03_astrf/log/mergescampcat_genlist-00139388.log
Running: ../DES_HOME/bin/mergeScampCats.py --input wrap_firstcut.wcl >& ./03_astrf/log/mergescampcat-00139388.log
Running: ../DES_HOME/bin/genList.py --input wrap_firstcut.wcl >& ./03_astrf/log/runscamp_genlist-00139388.log
Running: ../DES_HOME/bin/runScamp.py --input wrap_firstcut.wcl >& ./03_astrf/log/runscamp-00139388.log
...
```

Following execution, the results of the run are available in more .wcl files, which contain information about what was run, what outputs were produced (by name and location), and what key meta-data parameters (quality of the some parameter, for instance, I.e. a residual error on photometric redshift) are also available:

Partial listing of an output.wcl file with provenance from running an astrometric code module (E. Bertin's SCAMP) on a DES exposure:

```
<wrapper>
   exitstatus = 0
   pipeline = astrorefine
   pipever = 1.2.3
   walltime = 32.46
</wrapper>
<file>
   <scamp_head_scamp_head>
      filename = ./03_astrf/data/D00155348_r_28_r01p01_scamp.head,
      ./03_astrf/data/D00155348_r_35_r01p01_scamp.head
   </scamp_head_scamp_head>
   <scamp_astrefcat>
      filename = UCAC-4
   </scamp_astrefcat>
 <req_metadata>
        equinox = 2000.0
        scampflg = 0
        crpix2 = 2048.0
        pv2_5 = 0.000517464328588
        cd2_1 = -7.27039176016e-05
        ctype2 = DEC--TPV
        ctype1 = RA---TPV
        cd2_2 = 1.89927362232e-07
        fwhm = 4.3086
        crpix1 = 2151.2
        pv2_8 = -0.00211133300424
```

What should we strive for?

- Desire loose coupling and clean APIs between components
- Historical information about making a coherent system
  - How the subsystems are hooked together and how they are utilized has been experiment specific.
  - Each experiment takes on a role of supporting and developing glue code (or integration code) that hold things together.
  - Experiments tend to take on one or more or parts of one area themselves to implement and support
  - Glue code is necessary: scripts that enable distributed and disparate analyses, record simulation results, track stored data, monitor jobs, and coordinate access to the conditions and calibrations
- Sharing components and expertise with other experiments is important
  - Of course LSST DM
  - Concepts and pieces of DES software
  - Facilities and work from the HEP community
  - Galaxy as a prototype user interface? (http://galaxyproject.org/)

Focus of our group at FNAL

- The applications and libraries, including the development and local execution
  - The APIs and wrapper protocols
  - Helping to design and provide analysis-specific tools
  - Helping to identify common tools across analysis applications
- The shared software frameworks and libraries
  - art framework: mu2e, g-2, LBNE/uBooNE (LArSoft), NOvA, DarkSide-50 (online)
  - Integral part of the design & development of CMSSW
- Computational Cosmology analysis portal (under development using Galaxy tools and NERSC resources)
- LSST Level-3 Science Analysis Toolkit: requirements, specifications, and prototypes

Conclusion

- The systems and concepts outlined here are useful to the LSST DESC analysis framework
  - We should watch for good overlap in what the DESC will need and the systems we have experience with
  - We will need to integrate several technologies to make a useful product, picking useful pieces that can be integrated with other projects and systems we will hear about today.
  - Curating and providing timely access to large volumes of data requires expertise and dedicated facilities
- The DESC framework is a slice of the LSST Science Analysis Toolkit
  - The DESC holds many of the more complex use cases
  - Previously called out as needing more requirements and well defined interfaces, and that also seems necessary here

End

Definitions

- Workflow
  - A *workflow management system* is a computer system that manages and defines a series of tasks … to produce a final outcome … - Wikipedia.
  - General Concept of Workflow: "The term workflow is used in computer programming to capture and develop human to machine interaction. *Workflow software aims to provide end users with an easier way to orchestrate or describe complex processing* of data …, but without the need to understand computers or programming" - Wikipedia.
- Framework
  - The entire suite of software used for DESC analysis, including scripts and tools needed to management jobs and data
  - An application skeleton used to build DESC analysis programs
  - An analysis program build using such a skeleton
  - ** A body of code, libraries, and interfaces that permit the construction of specific applications and sharing of algorithms and other modules using common ways of describing data and configuration

Relevant white paper tasks

- H-1. End-to-end Simulator capable of running at small scales
  - Test a wide variety of systematics
  - A simple set of procedures and scripts will be developed that will aid in job submission.
  - provide data handling tools that will help organize results and temporary files.
  - Aggressive provenance tracking
- H-2. Requirements for software framework
  - Develop the requirements for the software framework based on the use patterns of the analysis group.
  - define a list of packages to be included in the framework, a set of dependencies, and guidance for usage by the scientists.
- H-3. Level 3 Software framework
  - enable the science WG's to assess the significance of the systematics under consideration. T
  - need a uniform, robust way to quantify the effects of systematics and the algorithms used to mitigate them.
  - a suite of two-point function estimators;
  - simple analytic covariance matrices;
  - a theory code such as camb; and a Fisher estimator of bias and errors.
- H-4. Repository for code and data
  - access to a tested set of routines that is tracked over time and tested.
  - Simulation code in repository, precompiled binaries available
  - Repository accessible to everyone in the collaboration
  - Data from simulations housed in data stores,
  - users can import both and operate on any data set with any set of code.
  - External data sets, such as WMAP, Planck, DES, will also be easy to import when conducting analyses.
  - documented repository with easily categorized modules for analysis, linked to dependent code and data.

Technologies applied to DES

Applications	Workflow	Data handling	Data storage
Wrappers	Condor	GridFTP	Enstore
DES DP	DES DP/DM	(Globus Online)	dCache
Validation	Unit running		
Sim generation	WCL		

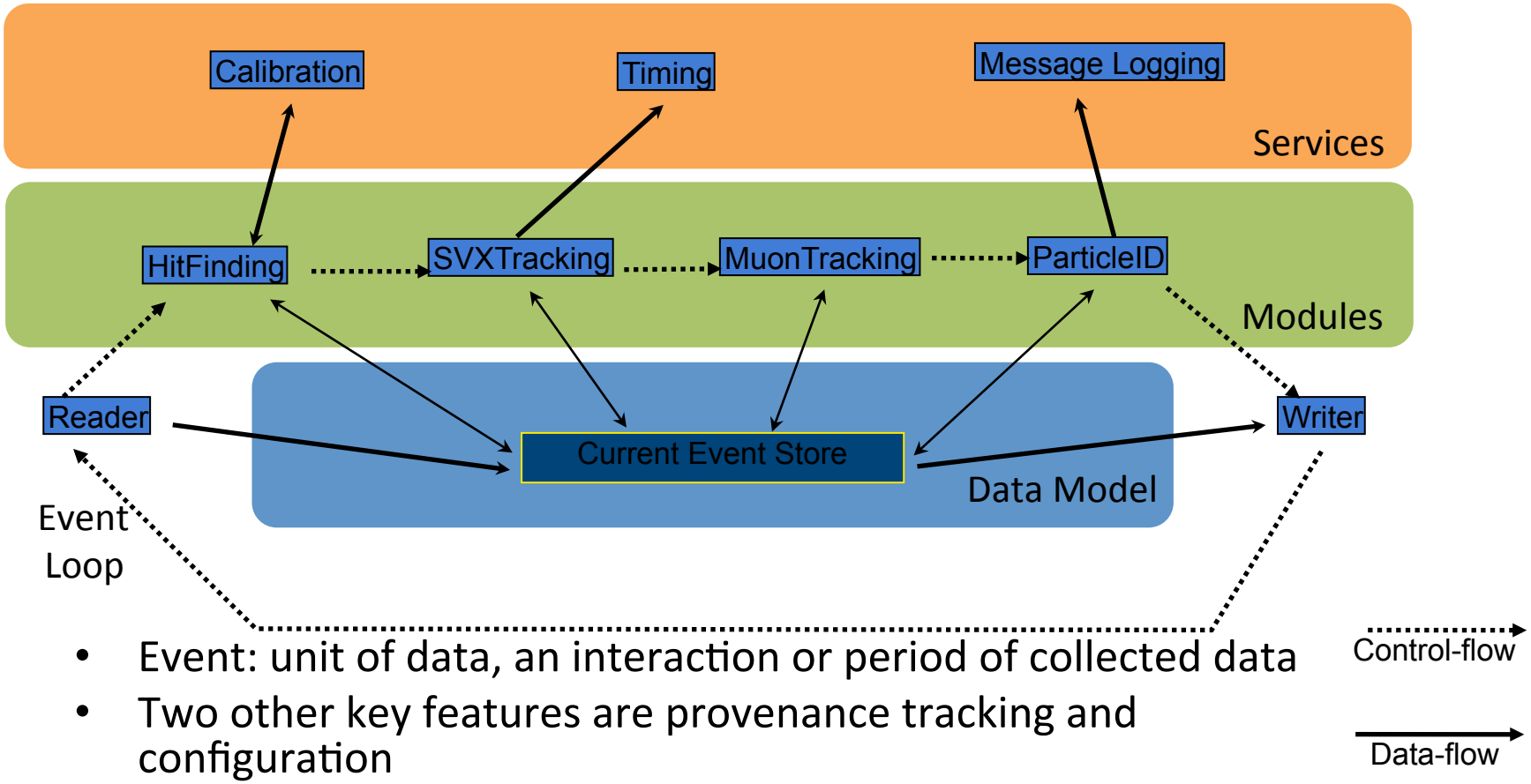
DES development and processing: … we focused on developing a Fermilab campus-based system for local DES processing, which augments the standard DES processing activity. Dark Energy Survey Dark Matter(DESDM) scientists, together with local Fermilab support and the OSG User Support Team, assembled a package for the Survey's November rush processing order. The DES Camera(DECam) obtains a set of about 300 mosaic (see Figure on left) camera images each night (each image is about 1 Gigabyte in size, consisting of 62 individual 2k x 4k CCD image files). In order to process each image, about 5GB of associated calibration and reference catalogs are required. The executable code is about 0.3 GB compressed.

FACS Tasks and Deliverables

- The pilot project proposed here will provide a software framework for analysis of survey data initially targeted for DES. The scope will be limited to parameter estimation problems needed by the Theory and Combined Probes Working Group involving MCMC. The software framework functionality to be delivered includes a collaborative hierarchical development environment to be used directly by physicists. The major pieces of this environment have already been descibed within the above design section. Key objectives of this project include ease of integration, deployment, and operation within existing and proposed cosmic frontier workflow systems and batch environments, along with use on individual scientist's workstations. Reuse of existing tools is essential. There are several existing tools that will be considered and used during construction of the project's subsystems, include standard software versioning and package management tools such as UPS from FNAL and modules available at NERSC, standard source code repository tools such as git and Mercurial, and standard build tools such as cmake and gnumake.
- Show picture of how this projects maps to the general parts.
- (this needs to be left out, for Scott to discuss. What is important is the type of framework application that this is and how it fits into the component model)

HEP: What we provide

Event Processing Framework: Software that coordinates the processing of collision events by pluggable reconstruction, filtering, and analysis modules. Events and modules are independent. Modules add data to and retrieve data from one event at a time.



- Event: unit of data, an interaction or period of collected data
- Two other key features are provenance tracking and configuration

Event Processing Framework

- Packaged as a toolkit with a set of software libraries
- Exists within the new relocatable UPS for easy of distribution, setup, and management